E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

EVALUASI FITUR *WORD2VEC* PADA SISTEM UJIAN ESAI *ONLINE*

Faisal Rahutomo [1)], Deasy Sandhya Elya Ikawati [2)], Obby Auliyaur Rohman [3)]
[1,2,3] Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
faisal@polinema.ac.id[1)], deasysandhya@polinema.ac.id[2)], obbyaur@gmail.com[3)]

ABSTRAK

Sampai sekarang pengembangan sistem informasi untuk mengevaluasi pemahaman siswa terhadap proses belajar memang sudah banyak dilakukan, baik berbentuk pilihan ganda maupun esai secara online. Dari referensi yang telah didapat kebanyakan menggunakan vektor kata dengan pendekatan skema pembobotan Term Frequency (TF). Dimana pada TF menghasilkan matriks kata yang senggang (Sparse), serta hasil error kemiripan yang besar. Didalam penelitian ini dievaluasi kinerja representasi fitur teks dan pembobotan yang lain, yaitu Word2vec. Selanjutnya perhitungan vektor kemiripan antar kedua teks menggunakan metode Cosine Similarity dan Euclidean Distance. Hasil evaluasi pengujian berupa nilai koefisien antara penilaian sistem dengan penilaian manual yang dibandingkan dengan penelitian terdahulu dengan skala sama. Data yang digunakan sebanyak 2162 data. Data ini diperoleh dari 50 siswa yang menjawab 40 soal (Politik, Olahraga, Lifestyle dan Teknologi). Hasil pengujian menunjukkan rata-rata nilai precentage error dengan menggunakan fitir Word2vec sebesar 59.5%. Angka tersebut menunjukkan nilai error yang tinggi. Sehingga penelitian ini sampai pada kesimpulan bahwa menggunakan fitur Word2vec tidaklah lebih akurat pada kasus ujian esai online dibanding menggunakan fitur Term Frequency (TF).

***Kata Kunci**: Cosine Similarity, Euclidean Distance, Text PreProcessing, Ujian Esai Online, Word2vec.*

ABSTRACT

Up to now, the development of information systems in evaluating students' understanding of the learning process has been carried out, both in the form of multiple choices and online essay. From the references that have been obtained, most related research use word vectors with the Term Frequency (TF) weighting scheme approach. Where the TF produces a free word matrix (Sparse), as well as the results of a large similarity error. In this study, the text feature performance and other weighting evaluations are evaluated, Word2vec. Furthermore, the similarity vector calculations between the two texts use the Cosine Similarity and Euclidean Distance methods. The testing evaluation results in the form of a coefficient value between the assessment of the system and manual assessment compared with previous studies with the same scale. The data used were 2162 data and were obtained from 50 students who answered 40 questions (Politics, Sports, Lifestyle and Technology). The test results show the average value of percentage errors using Word2vec's size of 59.5%. This number shows a high error value. Therefore, this study arrived at the conclusion that using the Word2vec feature is not more accurate in online essay exam cases than using the Term Frequency (TF) feature.

***Keywords**: Cosine Similarity, Euclidean Distance, Online Essay Exam, Text PreProcessing, Word2vec.*

I. PENDAHULUAN

Perkembangan teknologi informasi yang maju telah banyak membantu manusia dalam segala bidang. Salah satunya adalah bidang pendidikan. Setiap proses pembelajaran dalam Pendidikan tentunya memerlukan suatu proses evaluasi guna mengukur tingkat pemahaman siswa. Beberapa penelitian mengungkapkan bahwa evaluasi pembelajaran dalam bentuk esai lebih akurat dalam menentukan tingkat pemahaman siswa terkait pembelajaran yang di peroleh, dikarenakan siswa harus menuliskan informasi-informasi yang diketahui secara lebih rinci dalam menjawab pertanyaan yang diajukan[1].

Sampai sekarang, pengembangan sistem informasi untuk mengevaluasi pemahaman siswa terhadap proses belajar mengajar memang sudah banyak, baik dalam pilihan ganda maupun esai secara *online*. Namun untuk penilaian esai masih terus dikembangkan hingga sekarang, guna memperoleh hasil akurasi yang lebih baik dalam memberikan penilaian. Penelitian mengenai ujian esai berbahasa Indonesia telah banyak dilakukan untuk menentukan metode terbaik. Salah satunya yakni Analisis Aspek-Aspek Ujian Esai Daring Bebahasa Indonesia yang telah menggunakan perbandingan antara vector-vektor kemiripan antar kedua teks menggunakan metode *Cosine Similarity*, *Euclidean Distance*, dan *Jaccard* yang menghasilkan nilai *Percentage Error* dari *Cosine Simiarity* yakni 59.49%, *Euclidean Distance* 332.90%, dan *Jaccard* 52.31% [1]. Dimana penelitian tersebut menggunakan vektor kata dengan pendekatan skema pembobotan *Term Frequency* (*TF*). Dimana pada skema pembobotan *TF* menghasilkan

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

matriks kata yang senggang (*Sparse*), serta hasil *error* kemiripan yang besar. Terdapat banyak skema pembobotan yang ada, yang bisa digunakan untuk evaluasi dalam fitur ektraksi dan pembobotan teks.

Fitur ekstraksi dan pembobotan lain yang dapat digunakan untuk mengevaluasi penelitian yang ada yakni menggunakan *Word2vec*. *Word2vec* merupakan nama *word vector representation* yang dibuat oleh *Google*. Sebagai gambaran bahwa vektor dari *Word2vec* ini bisa mewakili makna dari sebuah kata, kita bisa mengukur beberapa vektor sebagai perbandingan. *Word2vec* dipilih karena menggunakan fitur matriks kata yang padat (*Dense Matrix)* dan dapat mereduksi dimensi menjadi lebih padat dari pada *Term Frequency (TF)* yang menggunakan *Sparse Matrix*, sehingga diharapkan dapat memberikan hasil yang lebih baik. Berdasarkan hal tersebut, penelitian ini ditujukan untuk mengetahui perbandingan menggunakan *Word2vec* dan *Term Frequency* (*TF*) pada skema kemiripan *Cosine Simiarity* dan *Euclidean Distance*.

II. LANDASAN TEORI

A. *Text PreProcessing*

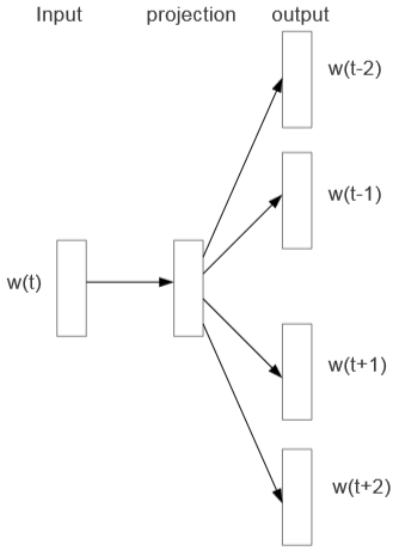
Pada data teks perlu adanya suatu proses tahapan yang bernama tahapan *preprocessing*, yakni mengubah data teks menjadi data numerik yang dapat diolah. Sehingga menjadikannya berbeda dengan pengolahan pada data numerik. Dalam *preprocessing* ada beberapa tahap yang harus dilakukan, yakni terdiri dari *Case Folding* (mengubah teks menjadi huruf kecil), *Tokenizing* (memecah teks menjadi kata), *Stemming* (mengubah kata menjadi kata dasar), dan *Stopword* (membuang kata yang tidak diperlukan) [2].

B. *Word2vec*

Word2Vec merupakan suatu representasi kata dalam bentuk vektor yang dibuat oleh *Google*. *Word2Vec* juga merupakan sekumpulan beberapa model yang saling berkaitan yang digunakan untuk menghasilkan *word embeddings*. *Word embeddings* merupakan sebutan dari seperangkat bahasa pemodelan dan teknik pembelajaran fitur atau *feature learning* pada *Natural Language Processing* (NLP) dimana setiap kata dari kosakata (*vocabulary*) memiliki vektor yang mewakili makna dari kata tersebut dan kata-kata tersebut dipetakan ke dalam bentuk vektor bilangan riil [3].

Word2vec menggunakan sekumpulan teks yang besar (*corpus*) sebagai data latih (*training*) untuk membangun *vocabulary* dan menghasilkan ruang vektor yang dapat berjumlah beberapa ratus dimensi, dengan setiap kata unik pada *corpus* berupa vektor dimana pembentukan vektor tersebut menerapkan model *Skip-gram* dan model CBOW (*Continous Bag-of-Words*) [4].

Arsitektur dari *Skip-Gram* ditunjukkan pada Gambar 3, yaitu terdiri atas sebuah lapisan tersembunyi *Neural Network.* Pembentukan model *Skip-Gram* dipengaruhi oleh beberapa parameter, di antaranya jumlah dimensi dari neuron (yang disebut dengan variabel *d*) dan jumlah *window* kata (yang disebut dengan variabel *t*) yang digunakan.



Gambar. 1. Ilustrasi model *Neural Network* untuk *Word2Vec Skip-Gram* [5]

Jika sebuah dokumen dilambangkan dengan *D* dan setiap kata yang ada di dalam dokumen tersebut dilambangkan dengan *wi* dengan *i* merupakan indeks kata di dalam dokumen, dan *T* adalah jumlah kata yang ada

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

di dalam dokumen, maka sebuah dokumen dapat dimodelkan dengan *D*={*w1*,*w2*,*w3*,...,*wT*}. Proses pelatihan *Neural Network* dari model *Skip-Gram* jika diberikan nilai window *c* = 3, maka untuk setiap *wi* akan digunakan sebagai masukan dari *Neural Network* untuk menebak sekumpulan kata dari *wc-i*,...,*wc+i*, yaitu *wi-3*,...,*wi+3*.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p\left(w_{t+j}|w_t\right) \quad (1)$$

Di mana *c* adalah ukuran konteks (*window*) data pelatihan (*training*) yang bisa menjadi fungsi dari pusat dari kata *wt*. Lebih besar nilai *c* maka lebih banyak pula hasil dari nilai training dan karenanya dapat menghasilkan akurasi yang lebih tinggi, dengan kosekueinsi waktu yang lebih lama.

$$p\left(w_j|w_i\right) = \frac{e^{\left(v'_{wj}{}^{T} v_{wi}\right)}}{\sum_{k=1}^{m} e^{\left(v'_{wk}{}^{T} v_{wi}\right)}} \quad (2)$$

Dimana variabel *vw* dan *v'w* adalah representasi vektor masukan dan keluaran dari sebuah kata *w*. Variabel *m* adalah jumlah kata unik yang ada di daftar kosakata. *One-hot encoding* digunakan sebagai representasi kata dari masukan dan keluaran menjadi sebuah vektor.

Model hasil pelatihan *Neural Network* adalah sebuah matriks *Memb* dengan ukuran sebesar *m x d*. Variabel *d* adalah jumlah dimensi vektor kata dari *Word2vec*. Untuk mendapatkan sebuah vektor dari kata, vektor *one-hot vw* dari sebuah kata *w* dikalikan dengan matriks *Memb*. Persamaan (3) merupakan fungsi variable $v_{embedding}$.

$$v_{embedding} = v_w{}^{T} * M_{emb.} \quad (3)$$

Persamaan untuk mendapatkan vektor dokumen dituliskan melalui (4).

$$emb_i = \frac{\sum_{j=1}^{n} v_j^i}{n} \quad (4)$$

Variabel *embi* adalah nilai vektor dokumen untuk dimensi ke-*i*. Variabel *n* bernilai jumlah kata yang ada di dalam dokumen. *vij* adalah isi elemen ke-*i* dari representasi vektor kata ke-*j*. Dari (4) tersebut didapatkan sebuah vektor dokumen *vdoc* = [*emb1, emb2, emb3*, …, *embd*] sebagai model representasi dari masing-masing dokumen jawaban dan kunci jawaban.

C. *Cosine Similarity*

Penentuan kesesuaian kunci jawaban dengan jawaban dipandang sebagai pengukuran (*similarity measure*) antara vektor kunci jawaban (D) dengan vektor jawaban (Q). Semakin sama suatu vektor kunci jawaban dengan vektor jawaban maka kunci jawaban dapat dipandang semakin sesuai dengan jawaban. Persamaan (5) digunakan untuk menghitung *Cosine Similarity*.

$$\text{CosSim (d,q)} = \frac{d\ .\ q}{||d||\ ||q||} \quad (5)$$

D. *Euclidean Distance*

Euclidean Distance merupakan rumus untuk mencari jarak dari 2 titik yang paling umum digunakan. *Euclidean Distance* masuk kedalam kategori perhitungan *dissimilarities*, koefisien yang dihasilkan menunjukkan perbedaan titik, sehingga koefisien perlu dikurangi koefisien maksimal dari perbedaan paling jauh 2 titik. Kemiripan dengan *Euclidean Distance* dua buah vektor didefinisikan seperti persamaan (6):

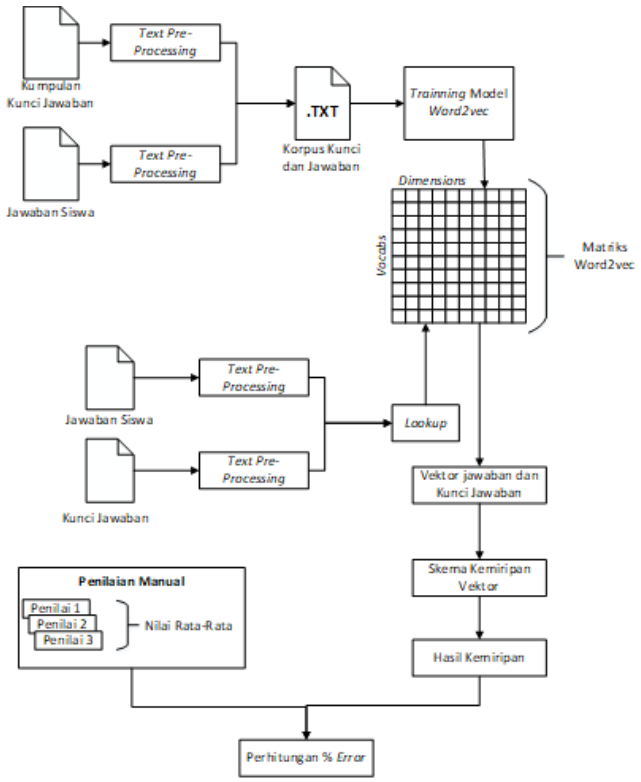
$$\textit{Euqlidean}\ (q, d) = \sqrt{\sum_{k=1}^{t}(q_k - d_k)^2} \quad (6)$$

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

III. METODOLOGI

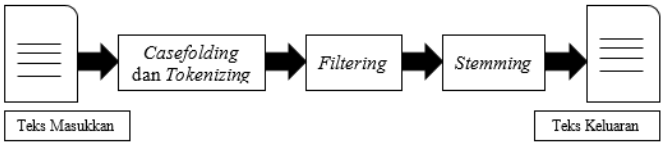
Pada penelitian ini, sistem dibagi menjadi beberapa proses, pertama yaitu melakukan proses *Text PreProcessing*, kemudian melakukan proses representasi vektor kata, dilanjutkan dengan mencari kemiripan antara vektor-vektor kata tersebut, dan terakhir mencari *error* dengan menggunakan *precentage error*.



Gambar. 2. Rancangan Penelitian pada Sistem

A. *Text PreProcessing*

Pada tahap ini teks dari kunci jawaban dan jawaban siswa perlu dilakukan proses *pre-processing* sebelum teks tersebut di ekstraksi menjadi data numerik yang dapat diolah, dikarenakan supaya data teks tersebut agar lebih bersih dari kata-kata yang tidak diperlukan yang dapat dilihat pad gambar 3.



Gambar. 3. Proses *Text PreProcessing*

B. *Word2vec*

Langkah pertama yakni mengambil data semua kunci jawaban dan jawaban siswa menggunakan *querry* dari *database*, kemudian hasil dari *querry* tersebut di text *preprocessing* dan dibuat menjadi file teks "corpus.txt". Kemudian file tersebut di proses menjadi model word2vec dengan menggunakan *library Gensim* yang ada pada bahasa pemrograman *Python*. Proses pembuatan model *word2vec* disebut *trainning*, dimana pada proses *trainning* tersebut akan menghasilkan matriks antar kata yang memiliki dimensi *m x d*. Dimana *m* merupakan jumlah banyak *vocab* atau kata atau *term* yang ada dan d merupakan jumlah *features* atau dimensi vektor. Tiap-tiap kata atau *vocab* akan direpresentasikan menjadi 800 dimensi vektor. Besar dimensi tersebut dipilih karena hasil perhitungan yang dihasilkan nantinya lebih sesuai dan mendekati terhadap hasil dari penilaian manual dibandingkan jumlah dimensi yang lain. Proses *trainning* juga menggunakan algoritma *Skip-gram* sebagai algoritma pembuatan model.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

1) Data Preparation

Sebagai contoh, digunakan data korpus yang berisis 5 kata/*vocab*, yaitu "poleteknik negeri malang selalu dihati". Dimana pada korpus tersebut digunakan parameter *d* = 10 dan *c* = 3. Kemudian dari korpus tersebut dilakuka *one-hot vector encoding*, yaitu mengubah kata target menjadi 1 dan lainnya menjadi 0.

#	Token	#1			#2				#3					#4					#5		
0	politeknik	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	negeri	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
2	malang	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0
3	selalu	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1
4	dihati	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0
		Xk	Y(c=+1)	Y(c=+2)	Xk	Y(c=-1)	Y(c=+1)	Y(c=+2)	Xk	Y(c=-2)	Y(c=-1)	Y(c=+1)	Y(c=+2)	Xk	Y(c=-2)	Y(c=-1)	Y(c=+1)	Y(c=+2)	Xk	Y(c=-2)	Y(c=-1)

Gambar. 4. Data *Preparation*

Jadi, dalam proses perhitungan antinya akan dilakukan sebanyak 5 kali, atau berdasarkan jumlah *m*. kemudian juga terdapat *learning rate* dan *epoch* atau pengulangan.

2) Calculate Hiden Layer & y_pred

Proses pembobotan pada *word2vec* berada pada *hidden layer*, terdapat dua pembobotan yaitu *W1* dan *W2*. Dimana pada *default word2vec*, kedua bobot tersebut diinisialisasikan sebagai *random number* sesuai jumlah *m* x *d*. Namun pada perhitungan ini *W1* dan *W2* diinisialisasikan antara -1 sampai 1. Pertama yaitu mencari *h* atau *hidden layer* dari suatu *wt*.

#	Token	Input - w_t		Weight 1 - W1										Hidden Layer - h
0	politeknik	1		0.236	-0.962	0.686	0.785	-0.454	-0.833	-0.744	0.677	-0.427	-0.066	0.236
1	negeri	0		-0.907	0.894	0.225	0.673	-0.579	-0.428	0.685	0.973	-0.070	-0.811	-0.962
2	malang	0		-0.576	0.658	-0.582	-0.112	0.662	0.051	-0.401	-0.921	-0.158	0.529	= 0.686
3	selalu	0		0.517	0.436	0.092	-0.835	-0.444	-0.905	0.879	0.303	0.332	-0.275	0.785
4	dihati	0	np.dot	0.859	-0.890	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237	-0.454
		1 x 5		5 x 10										-0.833
														-0.744
														0.677
														-0.427
														-0.066
														1 x 10

Gambar. 5. *Calculate Hidden Layer*

Kemudian menghitung *output layer*, yaitu hasil perkalian dari *h* dengan *W2*.

0.236		-0.868	-0.406	-0.288	-0.016	-0.560	
-0.962		-0.395	0.890	0.685	-0.329	0.218	Output Layer
0.686		-0.128	0.685	-0.828	0.709	-0.420	1.258
0.785		0.881	0.238	0.018	0.622	0.936	-1.369
-0.454	np.dot	-0.478	0.240	0.820	-0.731	0.260	= -1.828
-0.833		0.679	0.721	-0.111	0.083	-0.738	1.196
-0.744		-0.690	0.907	0.464	-0.022	-0.005	0.545
0.677		-0.054	0.397	-0.017	-0.563	-0.551	1 x 5
-0.427		-0.838	0.053	-0.160	-0.164	-0.671	
-0.066		0.096	-0.995	-0.313	0.881	-0.402	
1 x 10		10 x 5					

Gambar. 6. *Output Layer*

Dari hasil *output layer* kemudian di hitung nilai *softmax*, sehingga diperoleh *y_pred*.

#	Token	Output Layer	Softmax	
			Unnormalized Probabilities	Normalized Probabilities
0	politeknik	1.258	3.517	0.392
1	negeri	-1.369	0.254	0.028
2	malang	-1.828	0.161	0.018
3	selalu	1.196	3.308	0.369
4	dihati	0.545	1.724	0.192

Gambar. 7. Hasil *y_pred*

3) Mencari Error dan SUM of Different

Selanjutnya mencari jumlah *SUM of Different* pada *wt* atau $\sum I$. Sebagai contoh pada kata "politeknik", mempunyai *y_pred* "negeri" dan "malang". Maka, pada kata "politeknik" akan dicari jumlah error yang diperoleh dari *y_pred* "negeri" dan "malang". Sedangkan untuk mencari *error* sendiri diperoleh darihasil pengurangan *one-hot encoding* dengan *softmax*.

y_pred Softmax	w_c = 1	Token	Diff	y_pred Softmax	w_c = 2	Token	Diff	EI Sum of Diff
0.392	0	politeknik	0.392	0.392	0	politeknik	0.392	0.785
0.028	1	negeri	-0.972	0.028	0	negeri	0.028	-0.943
0.018	0	malang	0.018	0.018	1	malang	-0.982	-0.964
0.369	0	selalu	0.369	0.369	0	selalu	0.369	0.738
0.192	0	dihati	0.192	0.192	0	dihati	0.192	0.385

Gambar. 8. Hasil *EI* untuk Kata "politeknik".

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

4) Backpropagation

Simpelnya proses ini bermaksud untuk menyesuaikan kembali tiap *weight* dan bias berdasarkan *error* yang didapatkan. Pada tahap ini akan dicari nilai *delta* dari masing-masing *W*. Untuk delta pada *W2* didapatkan dengan perkalian *h* dan $\sum I$.

Hidden Layer		EI		Delta for W2				
0.236		Sum of Diff		0.185	-0.222	-0.227	0.174	0.091
-0.962		0.785		-0.755	0.907	0.927	-0.710	-0.370
0.686		-0.943		0.538	-0.647	-0.661	0.506	0.264
0.785		-0.964		0.616	-0.740	-0.757	0.579	0.302
-0.454		0.738		-0.356	0.428	0.438	-0.335	-0.175
-0.833	np.outer	0.385	=	-0.654	0.786	0.803	-0.615	-0.321
-0.744				-0.584	0.702	0.718	-0.549	-0.286
0.677				0.531	-0.638	-0.652	0.499	0.260
-0.427				-0.335	0.402	0.411	-0.315	-0.164
-0.066				-0.051	0.062	0.063	-0.048	-0.025
10 x 1		5 x 1		10 x 5				

Gambar. 9. Hasil *Delta W2*.

Sedangkan untuk mencari *delta W1* sedikit lebih rumit, yakni pertama mencari *dot* dari $\sum I$ dan W2.

Weight (W2)						EI		np.dot(W2, EI)
-0.868	-0.406	-0.288	-0.016	-0.560		Sum of Diff		-0.247
-0.395	0.890	0.685	-0.329	0.218		0.785		-1.968
-0.128	0.685	-0.828	0.709	-0.420		-0.943		0.414
0.881	0.238	0.018	0.622	0.936		-0.964		1.269
-0.478	0.240	0.820	-0.731	0.260		0.738		-1.831
0.679	0.721	-0.111	0.083	-0.738	np.dot	0.385	=	-0.263
-0.690	0.907	0.464	-0.022	-0.005		5 x 1		-1.862
-0.054	0.397	-0.017	-0.563	-0.551				-1.028
-0.838	0.053	-0.160	-0.164	-0.671				-0.932
0.096	-0.995	-0.313	0.881	-0.402				1.812
10 x 5								10 x 1

Gambar. 10. Hasil *dot EI* dan *W2*.

Kemudian untuk membuat hasil *dot* tersebut menjadi *delta W1*, hasil *dot* tersebut dikalikan dengan *one-hot encoding* dari *wt*.

w_t		np.dot(W2, EI)		Delta for W1 np.outer(w_t, np.dot(W2, EI))									
1		-0.247		-0.247	-1.968	0.414	1.269	-1.831	-0.263	-1.862	-1.028	-0.932	1.812
0		-1.968		0	0	0	0	0	0	0	0	0	0
0	np.outer	0.414		0	0	0	0	0	0	0	0	0	0
0		1.269		0	0	0	0	0	0	0	0	0	0
0		-1.831	=	0	0	0	0	0	0	0	0	0	0
5 x 1		-0.263		5 x 10									
		-1.862											
		-1.028											
		-0.932											
		1.812											
		10 x 1											

Gambar. 11. Hasil *delta W1*.

5) Update Weight

Setelah diperoleh nilai delta dari masing-masing *weight*, selanjutnya meng-update nilai dari masing-masing weight dengan cara *Weight – learning rate* (0.025) * *Delta Weight*. Berikut merupakan *W* atau bobot dari kata "polinema".

Updated Weight (W1)									
0.24202	-0.91254	0.67541	0.75313	-0.40812	-0.82681	-0.69776	0.70245	-0.40323	-0.11084
-0.90686	0.89372	0.22517	0.67294	-0.57859	-0.42823	0.68545	0.97326	-0.06998	-0.81135
-0.57642	0.65794	-0.58247	-0.11187	0.66228	0.05126	-0.40076	-0.92055	-0.15796	0.52942
0.51692	0.43591	0.09182	-0.83478	-0.44396	-0.90529	0.87882	0.30314	0.33212	-0.27488
0.85921	-0.88984	0.65071	0.18497	-0.51058	-0.45623	0.37692	-0.27448	0.18150	-0.23724
5 x 10									

Gambar. 12. Hasil *update W1*.

Setelah diperoleh nilai *update W1* dan *W2*, kemudian dilakukan pengulangan atau *epoch* sebanyak yang diperlukan. Sehingga hasil akhir dari proses *trainning* diatas adalah nilai *m* x *d*. Jadi, berdasarkan proses diatas untuk kata "polinema" akan memiliki nilai vektor sebesar = 0.24202, -0.91254, 0.67541, 0.75313, -0.40812, -0.82681, -0.69776, 0.70245, -0.40323, -0.11084. Berikut merupakan hasil vektor dari korpus "politeknik negeri malang selalu dihati" tanpa menggunakan *epoch*.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

Updated Weight (W1)

0.24202	-0.91254	0.67541	0.75313	-0.40812	-0.82681	-0.69776	0.70245	-0.40323	-0.11084
-0.90387	0.84366	0.19208	0.68277	-0.60497	-0.44284	0.63357	0.94279	-0.08905	-0.73892
-0.57641	0.63999	-0.52603	-0.09920	0.62023	0.08395	-0.40613	-0.91862	-0.15511	0.54035
0.51196	0.43767	0.07729	-0.82800	-0.43500	-0.92139	0.87539	0.29780	0.32426	-0.28412
0.86333	-0.87875	0.62931	0.17138	-0.49104	-0.46212	0.38542	-0.26851	0.18676	-0.24116

5 x 10

Gambar. 12. Hasil vektor *m x d.*

C. Menghitung Nilai Kemiripan

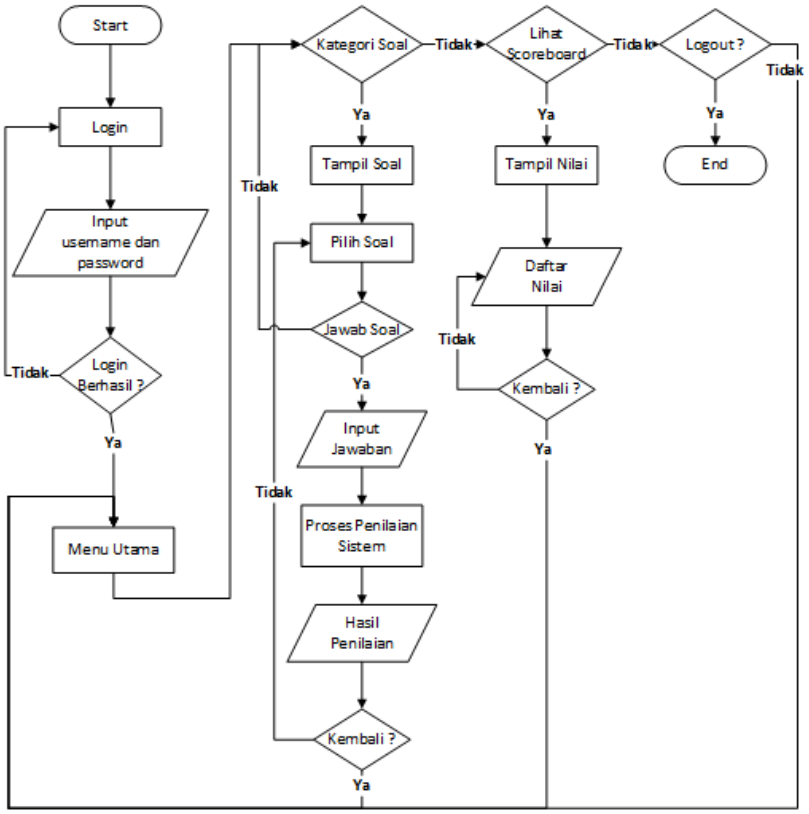
Setelah didapatkan vektor dokumen jawaban dan kunci jawaban, maka dilakukan perhitungan untuk mencari kemiripan antara kedua vektor dokumen tersebut. Tahapan perhitungan ini menggunakan metode *cosine similarity* dan *euclidean distance*. Nilai yang dihasilkan dari perhitungan tersebut merupakan kemiripan dari dua vektor dokumen. Hasil dari nilai kemiripan tersebut kemudian dihitung nilai *presentage error*-nya.

D. Implementasi

Implementasi adalah penulisan bahasa pemograman pada komputer, agar kode program dapat berjalan sesuai dengan rancangan.

1) Flowchart

Berikut gambar 4 menjelaskan *flowchart* dari sistem Evaluasi Fitur *Word2vec* pada Ujian Esai *Online*. Pertama melakukan proses login untuk masuk kedalam sistem, kemudian pada halaman bernda diberikan beberapa opsi. Opsi pertama yaitu untuk melihat daftar soal berdasarkan kategori dan menjawab soal tersebut dan melihat nilai yang diberikan sistem. Opsi kedua melihat daftar nilai dari soal-soal yang telah terjawab. Opsi terakhir yaitu logout.



Gambar. 4. *Flowchart* Sistem

2) Antarmuka Program

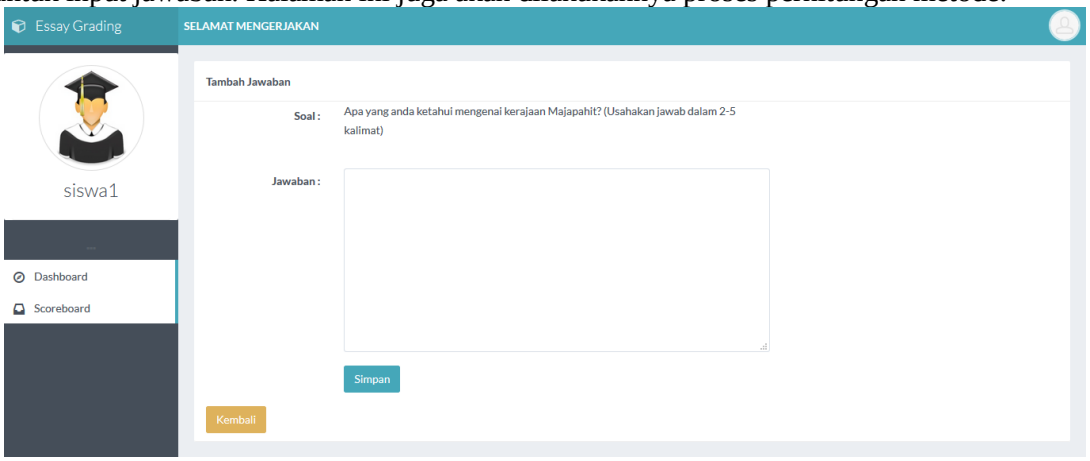
Antar muka digunakan untuk memudahkan *user* untuk menjawab soal dan melihat hasil perhitugan kemiripan. Pada antar muka ini akan dijabarkan di beberapa bagian yang akan digunakan oleh *user*.

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45

a. Halaman Menjawab Soal

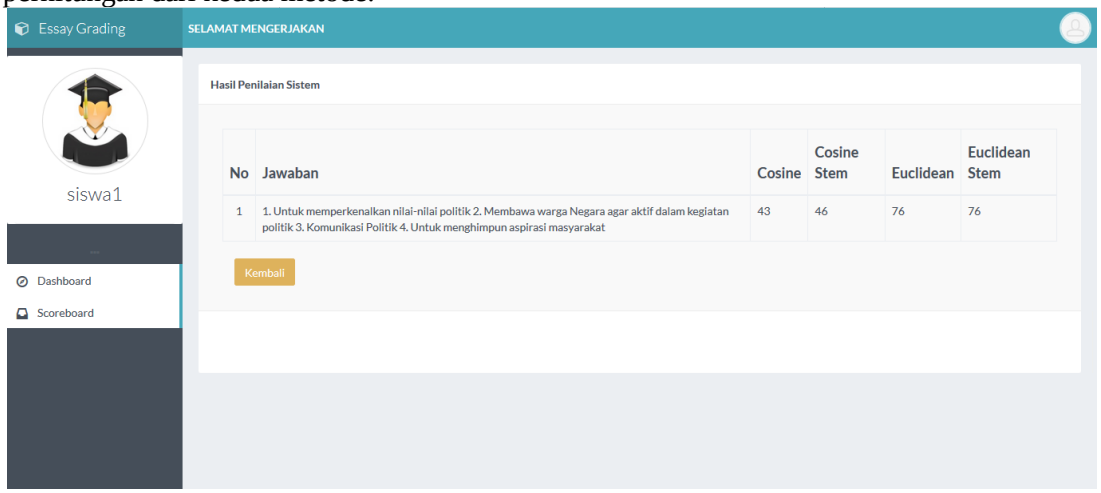
Halaman menjawab soal merupakan halaman yang menampilkan informasi soal yang dipilih dan *form* untuk input jawaban. Halaman ini juga akan dilakukannya proses perhitungan metode.



Gambar. 5. Form Menjawab Soal

b. Halaman Detail Jawaban

Halaman detail jawaban merupakan halaman yang menampilkan informasi penilaian oleh sistem terhadap jawaban dan kunci jawaban dari soal terkait. Halaman ini berisi detail jawaban dan nilai perhitungan dari kedua metode.



Gambar. 6. Form Detail Jawaban

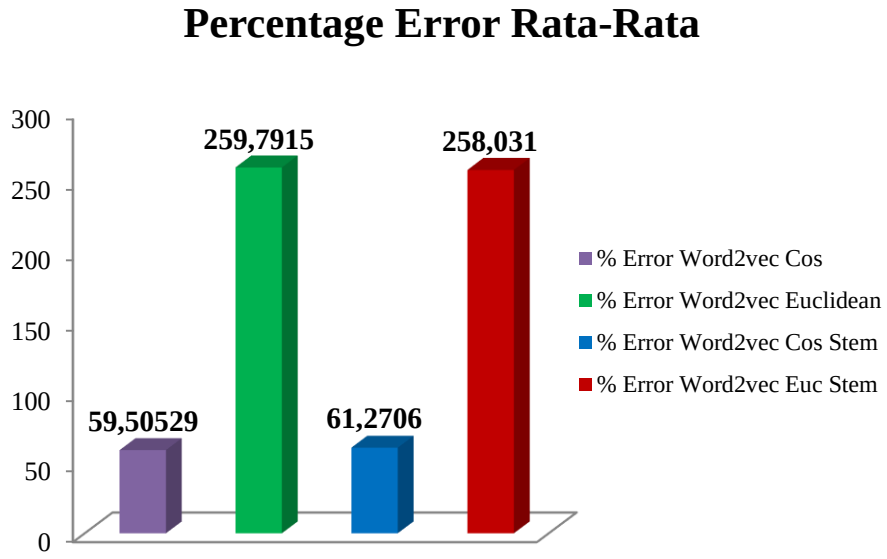
IV. PENGUJIAN DAN PEMBAHASAN

Pengujian sistem bertujuan untuk menghitung tingkat keakurasian dari hasil perhitungan oleh sistem yaitu menggunakan perhitungan *percentage error*, di mana dilakukan perhitungan perbandingan antara penilaian manual oleh manusia dengan penilaian sistem. Pada penelitian ini untuk menguji tingkat akurasi sistem digunakan data uji penelitian sebelumnya yaitu data uji ujian esai *online*. Di mana terdapat 2162 teks jawaban siswa yang sudah dinilai secara manual oleh seorang guru dan juga sudah dinilai secara sistem menggunakan metode-metode pada penelitian sebelumnya.

Pada penelitian ini pengujian akurasi dilakukan dengan cara menghitung nilai kemiripan teks antara kunci jawaban dan jawaban siswa. Kemudian mencari rata-rata absolut *precentage error* (*MAPE*) dari masing-masing metode yang digunakan. Semakin kecil nilai *error*, maka semakin akurat penilaian yang dilakukan sistem

E-ISSN : 2540 - 8984

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Volume 04, Nomor 01, Juni 2019 : 36 - 45



Gambar. 7. *Percentage Error* dari Semua Kategori.

Dari grafik pada Gambar 7, diketahui bahwa metode *Cosine Similarity* tanpa *stemming* memiliki nilai *percentage error* yang paling kecil bila dibandingkan metode yang lain, yaitu 59.51%. Sedangkan, *Cosine Similarity* dengan *stemming* memiliki *error* yang tidak jauh bebeda dari *Cosine Similarity* tanpa *stemming*, yaitu 61.27%. Perbedaan skema *Cosine Similarity* dengan *stemming* dan *Similarity* tanpa *stemming* sebesar 1.76%.

Pengujian yang lain yaitu membandingkan *MAPE* dari hasil menggunakan metode *word2vec* dengan metode lain pada penelitian sebelumnya. Tujuan dari pembandingan tersebut adalah untuk mengetahui apakah menggunakan vektor *Word2vec* lebih akurat dari metode-metode sebelumnya.

TABEL I
PERBANDINGAN RATA-RATA NILAI *PERCENTAGE ERROR*

Bidang	% Error Cosine	% Error Euclidean	% Error Jaccard	% Error Cos Stem	% Error Euc Stem	% Error Jacc Stem	% Error Word2vec Cos	% Error Word2vec Euclidean	% Error Word2vec Cos Stem	% Error Word2vec Euc Stem
Lifestyle	77.270	436.055	66.772	79.719	435.361	61.267	76.472	339.464	80.901	336.860
Politik	46.918	460.421	57.047	49.035	436.105	54.514	51.864	345.867	51.854	341.193
Olahraga	59.306	184.485	49.498	60.845	183.397	46.681	59.787	142.095	64.254	140.548
Teknologi	48.485	276.662	50.879	48.350	276.748	46.770	49.898	211.741	48.073	213.523
Rata-Rata	57.995	339.406	56.049	59.487	332.903	52.308	59.505	259.791	61.271	258.031
Peringkat	3	10	2	4	9	1	5	8	6	7

Dari keempat kategori pertanyaan, hasil nilai *percentage error* rata-rata keseluruhan dapat dilihat pada tabel 4 di atas. Dari tabel 1 tersebut dapat dilihat bahwa nilai *percentage error* menggunakan vektor *Word2vec* adalah 59.5% untuk *Cosine Similarity* dan menempatkan metode *Word2vec* pada peringkat kelima, itu artinya bahwa hipotesis penulis yang beranggapan bahwa dengan memanfaatkan vektor *Word2vec* tidak terbukti dapat meningkatkan nilai akurasi dalam menghitung penilaian soal esai *online*.

V. KESIMPULAN DAN SARAN

Dengan menerapkan fitur *Word2vec* pada teks jawaban dan kunci jawaban maka dapat dihasilkan sebuah matriks yang berisi vektor representasi dari tiap-tiap kata, dimana dapat dimanfaatkan untuk perhitungan kemiripan vektor. Penggunaan vektor dan pembobotan *Word2vec* untuk meningkatkan akurasi penilaian ujian esai *online* menghasilkan sebuah konklusi yang bersinggungan dengan hipotesis yang diharapkan penulis, karena setelah dibandingkan dengan hasil perhitungan metode-metode sebelumnya nilai *percentage error* metode *Word2vec* di-anggap masih cukup besar yaitu 59.5% dan tidak lebih baik atau akurat dengan metode-metode sebelumnya.

Text pre-processing yang dihasilkan dirasa kurang akurat untuk digunakan pada metode *Word2vec*, yakni di *stopword* atau *di stemming*, sehingga terdapat banyak kata yang mungkin penting yang dihilangkan. Terdapat

beberapa metode ekstraksi lain yang mungkin bisa digunakan, salah satunya adalah *Doc2vec*. Sama halnya dengan *Word2vec* yang mampu merepresentasikan teks menjadi vektor, namun *Doc2vec* bisa langsung merepresentasikan dokumen teks menjadi vektor, yang mungkin dapat menghasilkan perhitungan yang lebih akurat.

DAFTAR PUSTAKA

[1] Roshinta, Trisna Ari. "Analisa Aspek-Aspek Ujian Esai Daring Berbahasa Indonesia". Seminar Nasional Terapan Riset Inovatif Semarang, Volume 1. 2016.Denpasar Based on NASA Data," *J. Phys. Conf. Ser.*, vol. 953, no. 1, 2018.
[2] Manning, C. D., Raghavan, P., Schutze, H., (2009). An Introduction to Information Retrieval. Cambridge: Cambridge University Presss, 22-32.
[3] Kencana, I Kadek Bayu Arys Wisnu. "Klasifikasi Opini Pada Fitur Produk Berbasis Graph". *e-Proceeding of Engineering*, Vol.4, No.2, 2017.
[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *ICLR Workshop*, 2013.
[5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *Proceedings of the 27th Annual Conference on Neural Information Processing Systems* (NIPS), 2013.
[6] TensorFlow. "Vector Representations of Words". [Online]. Tersedia: https://www.tensorflow.org.
[7] M. Naili, A. H. Chaibi, dan H. H. Ben Ghezala, "Comparative Study of Word Embedding Methods in Topic Segmentation," Procedia Computer Science, Vol. 112, hal. 340–349, 2017.
[8] Carl J. Wenning, Ed.D. (2014). All Student Lab Handbook Physics Teacher. *Education Program Coordinator* 1994-2008, 12-32.
[9] http://flask.pocoo.org (Diakses pada 25/05/2019 06:48)
[10] http://flask.pocoo.org:80/docs/1.0/foreword (Diakses pada 25/05/2019 07:16)
[11] Rahutomo, Faisal, et al. "*Econo-ESA reduction scheme and the impact of its index matrix density*". Procedia Computer Science 35, 2014.
[12] Putra, Carfin Febriawan Pratama. "*Implementasi Explicit Semantic Analysis Berbahasa Indonesia Menggunakan Corpus Wikipedia Indonesia*". Jurnal Informatika Polinema, Vol4, Edisi4, 2018.
[13] Rahutomo, Faisal, et al. "*Open Problems in Indonesian Automatic Essay Scoring System*". International Journal of Engineering & Technology 44, 2018.
[14] Sari, Dhiana Novita. "Evaluasi Library Deep Learning - Keras Pada Ujian Esai Online". Politeknik Negeri Malang. 2019.
[15] Rahutomo, Faisal, et al. "*Semantic Cosine Similarity*". Graduate School of Science and Technology, Kumamoto University. 2012.
[16] Ali, Muhammad Haidar. "Evaluasi Hasil Implementasi Manhattan Distance Dan Dice Similiarity Pada Sistem Ujian Esai Daring Berbahasa Indonesia". Politeknik Negeri Malang. 2019.